1. 进程和线程区别

进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动。

线程是进程的一个实体。

程序是一组指令的集合。（进程=程序+资源）

1. 进程是操作系统进行资源~~调度和~~分配的基本单元，线程是CPU调度~~和分配~~的基本单元。
2. 一个线程必定属于也只能属于一个进程;而一个进程可以拥有多个线程并且至少拥有一个线程。
3. 属于同一进程的所有线程共享该进程的所有资源。包括打开的文件，套接字等。
4. 线程是轻量级进程，线程控制块比进程控制块小的多，所以线程间切换代价小，进程间切换代价大。
5. 每个进程有独立的内存空间，而线程共享所述进程的内存空间。
6. 线程同步方法
7. 临界区 ：通过对多线程的串行化来访问公共资源
8. 互斥量 ：实现对一个共享资源的单独访问，只有持有互斥量的线程才能才有权限去访问共享资源
9. 信号量：控制同一时刻访问资源的最大线程数
10. 事件：通知线程有一些事件已经发生，从而启动后继任务的开始
11. 进程通信方式
12. 共享内存
13. 管道
14. 消息队列
15. 事件
16. 虚拟内存

虚拟内存是计算机系统进行内存管理的一种技术，他使得应用程序认为它拥有连续可用的内存(一个连续完整的地址空间)，允许程序员编写比实际系统拥有的内存大得多的程序，这使得大型软件能够在有限内存资源的系统上实现。而实际上，它通常被分割为多个物理碎片，还有部分暂时存在外部存储器上，在需要时进行数据交换。

**优势：**

1. 扩大了地址空间。寻址空间比实际内存大
2. 内存保护。每个进程运行在各自的虚拟内存空间，互不干扰
3. 虚存共享实现进程间通信

**缺点：**

1. 页面的换入换出需要磁盘I/O,很耗时
2. 虚拟地址到物理地址转换，增加指令的执行时间
3. 虚存管理需要建立许多数据结构，占用额外的内存
4. 内存碎片

多次内存分配造成的，内存空白段小到无法给用户提供足够的空间。

**内碎片：分配给程序的空间没用完，**

**外碎片：**

1. 动态链接库和静态链接库
2. 死锁

**死锁**：是指两个或以上的进程在执行过程中，由于竞争资源或者由于彼此通信而造成的一种阻塞的现象，若无外力作用，他们将无法推进下去，此时称系统处于死锁状态。

(规范定义)集合中的每一个进程都在等待只能由本集合中的其他进程才能引发的事件，那么该组进程是阻塞的。

**死锁的四个必要条件：**

(1) 互斥：每次只有一个进程能使用某一特定资源。其他进程不能访问已分配给某一进程的资源。

(2) 占有且等待：当进程因为等待资源而阻塞时，继续占有已经分配的资源，不释放锁

(3) 不可抢占：不能强行抢占其他进程已占有的资源（例如当涉及数据库资源时，必须提供回滚机制以支持资源抢占，这样才能把进程和它的资源恢复到以前适当的状态，使得进程最终可以重复它的动作。）

(4) 循环等待：两个或两个以上的进程形成循环链，每个进程都在等待循环链中的另一个进程持有的资源。

**死锁预防：（设计一种系统，约束资源请求，防止4个死锁必要条件中至少一个的发生）**

(1) 互斥：做不到，写访问一般都是互斥的

(2) 占有且等待：为预防占有且等待的条件，可以要求进程一次性地请求所有需要的资源，并且阻塞这个进程直到所有请求都同时满足。(缺点：低效，事先不知道进程所需要的所有资源)

(3) 不可抢占：当占有某些资源的一个进程进行进一步资源请求被拒绝，则进程必须必须释放它最初占有的资源。

(4) 循环等待：定义资源类型的线性数据来预防。如果一个进程已经分配到R类型的资源，那么它接下来请求的资源只能是那些排在R类型之后的资源。

死锁避免：（允许四个必要条件，但通过明智的选择，确保永远不会到达死锁点，因此死锁避免比死锁预防允许更多的并发）

（1） 如果一个进程的请求会导致死锁，则不启动此进程

（2） 如果一个进程增加的资源请求会导致死锁，则不允许此分配